



# UNITED STATES PATENT AND TRADEMARK OFFICE

*MN*  
UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/637,453	08/08/2003	Benjamin E. Chelf	COV-1	3626
36532	7590	05/31/2007		
G. VICTOR TREYZ FLOOD BUILDING 870 MARKET STREET, SUITE 984 SAN FRANCISCO, CA 94102			EXAMINER KISS, ERIC B	
			ART UNIT	PAPER NUMBER
			2192	
			MAIL DATE	DELIVERY MODE
			05/31/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

# Office Action Summary

Application No.

10/637,453

Applicant(s)

CHELF ET AL.

Examiner

Eric B. Kiss

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 16 March 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-18 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 16 March 2007 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

### DETAILED ACTION

1. The reply filed March 16, 2007, has been received and entered. Claims 1-18 are pending.

#### *Drawings*

2. The drawings were received on March 16, 2007. These drawings are acceptable.

#### *Response to Arguments*

3. Applicant's arguments filed March 16, 2007, have been fully considered but they are not persuasive.

#### Claims 1-6

It is noted that applicant's specification does not set forth any conflicting definition of "build program" with reasonable clarity, deliberateness, and precision that would render the incorporation of such a definition into the claims appropriate. *See In re Paulsen*, 30 F.3d 1475, 1480, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994). Accordingly, the term "build program" has been broadly interpreted in light of the specification to mean any program that assists in the process of building executable code. The modified gcc compiler of Smatch, xgcc, is such a build program. Running the xgcc program of Smatch invokes a compiler (a modified gcc compiler) to compile code and produce .sm files.

The individual Smatch scripts may be considered static analysis tool management programs, which "invoke" static analysis tools (running the script programs invokes the functionality coded into the script, such as the functions described in *SmUsing*).

#### Claims 7-10

As applicant notes, the modified compiler in Smatch is called through the variable CC in the Makefile. In the Makefile, the variable CC normally refers to gcc (as accessible through the

Art Unit: 2192

unmodified search path). This definition is commented out and replaced with an alternate path to the xgcc compiler of Smatch. Thus, where an unmodified Makefile would invoke CC to compile code (where CC is mapped to “gcc” on the current search path), the modified makefile invokes CC now mapped to a different directory (essentially a search path consisting of only one directory, where the one directory is naturally first in the new search path) and invokes the modified xgcc compiler instead.

Note that the mapping of the modified xgcc compiler through the CC variable is giving the xgcc compiler the CC name, which is the same as CC name given to the gcc compiler in the unmodified Makefile.

#### Claims 11-14

It is noted that the output of Granston’s wrapper is not just the log file as applicant asserts, but rather the wrapper also invokes the compiler using the compiler command line. *See, e.g.,* Granston at col. 4, lines 12-15. Thus, Granston does teach using the output of the wrapper to run a build program, contrary to applicant’s characterization. Accordingly, the teachings of Granston, when combined with the modified gcc of Smatch, render claims 11-14 obvious.

#### Claims 15-18

While the examiner agrees that compilers are not operating systems, the examiner asserts that the compilers that may be invoked through operating system command lines may reasonably be considered operating system commands as recited in claims 15-18. Thus, the replacing of command-line-accessible gcc compiler with command-line-accessible xgcc compiler in Smatch is redefining an operating system command in the context of these claims.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., process creation and execution commands such as fork and exec) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

***Claim Rejections - 35 USC § 102***

4. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

5. Claims 1-10 and 15-18 are rejected under 35 U.S.C. 102(a) as being anticipated by the Smatch C source checker (hereinafter "*Smatch*"), as evidenced by the following documents (See MPEP § 2131.01):

- "Installing Smatch!!!", 10/11/2002 [online], accessed 12/19/2006, retrieved from Internet <URL:  
<<http://web.archive.org/web/20021011112904/http://smatch.sourceforge.net/installing.html>>  
(1 page) (hereinafter "*SmInst*");
- "Smatch!!!", 04/04/2003 [online], accessed 12/19/2006, retrieved from Internet <URL:  
<http://web.archive.org/web/20030404041020/http://smatch.sourceforge.net/>>, (3 pages)  
(hereinafter "*SmMain*");
- "Smatch Intermediate Code Representation!!!", 10/11/2002 [online], accessed 12/19/2006,  
retrieved from Internet <URL:

<http://web.archive.org/web/20021011071814/http://smatch.sourceforge.net/intermed.html>> (3 pages) (hereinafter “*SmIR*”);

- “Using Smatch!!!”, 04/13/2003 [online], accessed 12/19/2006, retrieved from Internet <URL: <http://web.archive.org/web/20030413091737/http://smatch.sourceforge.net/usingSmatch.html>>, (2 pages) (hereinafter “*SmUsing*”); and
- “Using smatch.pm!!!”, 07/15/2003 [online], accessed 12/19/2006, retrieved from Internet <URL: <http://web.archive.org/web/20030715162055/http://smatch.sourceforge.net/coding.html>> (5 pages) (hereinafter “*SmUpm*”).

As per claim 1, the *Smatch* documents disclose:

running a build program on the computer system to invoke compilers on the computer system that compile the source code files into executable code (Smatch uses a modified gcc compiler; see, e.g., *SmMain* at p. 1), wherein running the build program produces a build program output (*Id.*; the modified gcc compiler compiles the code, and also produces .sm files); and

running a static analysis tool management program on the computer system to invoke the static analysis tools and produce corresponding static error analysis results (see, e.g., *SmIR*, describing running a checker script on a bunch of .c.sm files generated by the compiler), wherein the static analysis tool management program accepts the source code and the build program output as inputs (note that the .sm files are representations of the source code).

As per claim 2, the *Smatch* documents further disclose directing the build program output to a file that is used as an input by the static analysis tool management program (the .c.sm files are piped through individual *Smatch* scripts; see, e.g., *SmMain* at p. 1).

As per claim 3, the *Smatch* documents further disclose directing the build program output to a file that is used as an input by the static analysis tool management program (the .c.sm files are piped through individual *Smatch* scripts; see, e.g., *SmMain* at p. 1), wherein the file contains information on which static analysis tools to substitute for each compiler when the static analysis tools are invoked (the individual *Smatch* scripts specify particular static analyses; see, e.g., *SmMain* at p. 1).

As per claims 4 and 5, the *Smatch* documents further disclose providing a user with an opportunity to specify for the static analysis tool management program which compiler options should be ignored and which additional compiler options are required by the static analysis tools when performing static analysis on the source code (see, e.g., *SmUsing* at p. 1; *Smatch* uses a modified gcc compiler that has an additional command-line option, --smatch; the user has the opportunity to interact with the modified compiler through the command line interface).

As per claim 6, the *Smatch* documents further disclose invoking a plurality of build management utilities with the build program as the build program is run, wherein the build program output includes output from the build management utilities (see, e.g., *SmUsing* at p. 1, describing the use of a Makefile).

As per claim 7, the *Smatch* documents disclose:

creating a new directory on a computer system (see, e.g., *SmInst* at p. 1);

modifying a search path on the computer system so that the new directory is included first in the search path (see, e.g., *SmUsing* at p. 1, describing the modification of the Make CC variable);

placing the static analysis tools into the new directory, wherein the static analysis tools in the new directory are given names matching the compiler names (see, e.g., *SmUsing*; the modified compiler is referenced by the CC variable in the Make configuration); and

running the build program so that the static analysis tools with the names matching the compiler names are invoked (the modified compiler is used to compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1).

As per claim 8, the *Smatch* documents further disclose:

obtaining information from a user on compilation options for the compilers (see, e.g., *SmUsing* at p. 1; *Smatch* uses a modified gcc compiler that has an additional command-line option, --smatch; the user has the opportunity to interact with the modified compiler through the command line interface); and

using the information on the compilation options when invoking the static analysis tools by running the build program (*Id.*).

As per claim 9, the *Smatch* documents further disclose running the build program comprising making calls to the compiler names (the modified compiler is used to compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1).

As per claim 10, the *Smatch* documents further disclose running the build program invoking both the compilers and the static analysis tools (the modified compiler is used to



Art Unit: 2192

compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1; the resulting .sm files are then processed by the *Smatch* scripts; see, e.g., *SmMain* at p. 1).

As per claim 15, the *Smatch* documents disclose:

redefining operating system commands in the operating system (the modified compiler is used to compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1); and

running the build program on the computer system, wherein the redefined operating system commands cause the build program to invoke the static analysis tools in place of the compilers so that the error analysis on the source code files is performed (the modified compiler is used to compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1; the resulting .sm files are then processed by the *Smatch* scripts; see, e.g., *SmMain* at p. 1).

As per claim 16, the *Smatch* documents further disclose using user-specified information on the compilers and compiler options during invocation of the static analysis tools (see, e.g., *SmUsing* at p. 1; *Smatch* uses a modified gcc compiler that has an additional command-line option, --smatch; the user has the opportunity to interact with the modified compiler through the command line interface).

As per claim 17, the *Smatch* documents further disclose redefining the operating system commands comprising redefining operating system process creation and execution commands by placing modified versions of the operating system creation and execution commands on the computer system and by instructing the operating system to load the modified versions of the operating system process creation and execution commands (see, e.g., *SmUsing* at p. 1).

As per claim 18, the *Smatch* documents further disclose redefining the operating system commands comprises using a new kernel module containing modified functions (see, e.g., *SmUsing* at p. 1 (step 1b)).

***Claim Rejections - 35 USC § 103***

6. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

7. Claims 11-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,960,202 (Granston et al.) and the *Smatch* documents (cited above in the rejection of claims 1-10 and 15-18 under 35 U.S.C. § 102(a)).

As per claim 11, *Granston et al.* discloses:

running a build program on a computer system (see, e.g., *Granston et al.* at col. 3, lines 59-61);

running a monitoring program on the computer system while the build program is running to compile the source code files (see, e.g. *Granston et al.* at col. 3, line 59, through col. 4, line 11), wherein the monitoring program monitors activity between the build program and the operating system (see, e.g. *Granston et al.* at col. 3, line 59, through col. 4, line 11).

*Granston et al.* fails to expressly disclose using output from the monitoring program to run the build program with static analysis tools substituted for the compilers so that the static analysis tools perform static error analysis on the source code files. However, the *Smatch* documents teach a system where a specialized compiler enabling static analysis is substituted for a normal compiler (the modified compiler is used to compile code for use with *Smatch*; see, e.g., *SmIR* at p. 1; the resulting .sm files are then processed by the *Smatch* scripts; see, e.g., *SmMain*

Art Unit: 2192

at p. 1). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to incorporate such a compiler substitution for static analysis as per the teachings of the *Smatch* documents. One would be motivated to do so to gain the advantages of enhanced error checking in source code compilation (see, e.g., *SmMain* at p. 1).

As per claim 12, *Granston et al.* further discloses gathering information from a user as to which compilers are used during the build process and which compilation options are to be used (see, e.g. *Granston et al.* at col. 3, line 59, through col. 4, line 11). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 13, *Granston et al.* further discloses filtering the output from the monitoring program to remove compiler option commands (see, e.g., *Granston et al.* at col. 2, lines 37-49). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 14, *Granston et al.* further discloses running the monitoring program comprises running a custom monitoring program that uses operating system debugging commands (e.g., generating a log file) to monitor the activity between the build program and the operating system (see, e.g. *Granston et al.* at col. 3, line 59, through col. 4, line 11). Therefore, for reasons stated above, such a claim also would have been obvious.

*Conclusion*

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist:  
571-272-2100.

A handwritten signature in black ink, appearing to read "Eric B. Kiss". The signature is stylized with a large "E" and "K".

Eric B. Kiss  
May 25, 2007